IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 October 2024, accepted 14 October 2024, date of publication 17 October 2024, date of current version 11 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3482850

RESEARCH ARTICLE

DC-DOES: A Dual-Camera Deep Learning Approach for Robust Orientation Estimation in Maritime Environments

FABIANA DI CIACCIO[1], SALVATORE TROISI[2], AND PAOLO RUSSO[3]
[1]Department of Civil and Environmental Engineering, University of Florence, 50139 Florence, Italy
[2]Department of Science and Technology, Parthenope University of Naples, 80143 Naples, Italy
[3]Department of Computer, Control and Management Engineering "Antonio Ruberti," University of Rome La Sapienza, 00185 Rome, Italy

Corresponding author: Paolo Russo (paolo.russo@diag.uniroma1.it)

ABSTRACT Attitude and Heading Reference Systems (AHRS) have achieved significant accuracy and reliability, making them suitable for various applications. This is possible through the integration of high-rate measurements, though they remain prone to errors, particularly sensor drift over time. As a potential solution, AHRS can be combined with complementary devices, such as camera-based systems, which have attracted attention for their cost-effectiveness and simplicity. This study introduces the Double Camera - Deep Orientation (roll and pitch) Estimation at Sea (DC-DOES), a Deep Learning model developed to enhance roll and pitch estimations obtained from conventional AHRS at sea. In comparison to previous versions, DC-DOES operates in a novel configuration utilizing a double-camera system. This system is based on a Jetson Nano embedded platform, integrating a low-cost AHRS and two synchronized cameras, resulting in a fully customizable acquisition and processing setup. DC-DOES is trained and validated on shore to assess its effectiveness and robustness in controlled conditions and will be further deployed on board for real-time applications at sea. It is trained on the Double Camera - ROll and PItch at Sea (DC-ROPIS) dataset, which was specifically collected for this purpose. Both the code and the dataset have been made publicly available to encourage further use and improvement. The results are promising, achieving a Mean Absolute Error (MAE) of approximately 1°, highlighting the potential of this cost-effective, reliable solution for orientation estimation tasks. Additionally, tests in low-light scenarios demonstrated its robustness under challenging conditions, making DC-DOES a suitable solution for maritime navigation and beyond.

INDEX TERMS AHRS, dataset acquisition, double-camera, deep learning, embedded systems, orientation estimation.

I. INTRODUCTION

The problem of pose estimation, i.e., determining the position and orientation of a vehicle, device, or human, is currently addressed using a variety of sensors, either integrated or stand-alone. Recent advancements across multiple fields—from terrestrial, maritime, and aerial navigation to human motion tracking and virtual reality—have increased the demand for more accurate orientation estimation.

While position can be reliably determined using Global Navigation Satellite Systems (GNSS), certain scenarios require alternative solutions [1], [2]. These latter often integrate techniques utilizing inertial sensors, odometry, laser, and sonar ranging sensors [3], combined with underwater positioning systems where necessary. The maritime environment, particularly in open-sea navigation, congested harbors, and waterways, presents a challenging setting where precise and accurate orientation data is crucial. This is also true for robotic navigation, both surface and underwater, especially considering the widespread use of low-cost, resource-constrained embedded systems in oceanographic

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Fadda.

research and environmental monitoring [4]. Here, research strategies and technological developments focus on achieving high mission productivity while minimizing operational costs.

Micro Electro-Mechanical Systems (MEMS)-based Attitude Heading Reference Systems (AHRS) integrate magnetometers with accelerometers and gyroscopes, enhancing orientation estimation in Inertial Measurement Units (IMU). These systems offer reduced size, weight, and cost. However, the low-cost nature of MEMS technologies introduces challenges that can affect the accuracy of pose estimation, necessitating both reliable and accurate solutions for optimizing localization and improving overall operational performance.

A cost-effective and easy-to-implement alternative is to utilize cameras to detect visual features, with significant advancements in visual-based techniques, such as Visual Odometry (VO) [5] and Visual Simultaneous Localization and Mapping (VSLAM) [6]. However, non-textured environments or poor lighting conditions remain challenges that the integration of IMU and camera systems—Visual Inertial Odometry (VIO) techniques—can only partially solve, even with precise parameter tuning [7].

Ongoing research in attitude estimation has benefited from the strong results produced by Deep Learning (DL) techniques, specifically through Deep Neural Networks. These models demonstrate robustness to camera parameters and challenging environments, making them a valuable tool for strengthening integrated systems [8].

Particularly relevant are the advancements in embedded technologies, where microprocessors and microcontrollers, functioning as small computers, are capable of performing real-time tasks. These devices, often categorized by processing power, cost, and architecture, have become integral components of small-scale robots. Recent technological developments have enabled even these small systems to run Deep Learning algorithms in real time [9].

In this context, this paper presents Double Camera - Deep Orientation Estimation at Sea (DC-DOES), a Deep Learning model designed to enhance orientation estimation in maritime navigation. DC-DOES builds upon the previous model [10], improving performance by incorporating a dual-camera setup and a low-cost AHRS sensor integrated into a fully customizable embedded Linux-based device. Moreover, the novel dataset Double Camera - ROll and PItch at Sea (DC-ROPIS), which contains paired images and corresponding orientation ground truth, was collected specifically for this purpose.

The approach aims to improve robustness in traditional methods under specific conditions, such as GNSS signal unavailability or long-lasting outages that cause significant drift in inertial sensors. It also addresses potential confusion with nearby robots equipped with SONAR or RADAR systems. However, DC-DOES is not intended to replace current systems but rather to complement them. Once deployed, DC-DOES relies entirely on visual features, making it immune to sensor drift over time. During deployment, the system architecture does not depend on AHRS, which is only used during the training phase.

The main contributions of this work are summarized as follows:

- A dual-camera system that captures two synchronized images of the horizon with perpendicular views.
- The development of a new embedded system for data acquisition and processing, based on the Nvidia Jetson Nano and a low-cost AHRS.
- The collection of a specific dataset reflecting the system's improvements and changes.
- Extensive testing of different Deep Learning model architectures, considering recent advancements to select the best-performing model for the task.

The rest of paper is organized as follows: Section II provides an overview of the existing literature on orientation estimation using inertial- and vision- based (included DL) methods. Section III introduces theoretical concepts related to attitude estimation and describes the tested Deep Learning architectures. Section IV details the integrated systems used to collect the DC-ROPIS dataset and describes the sensors and the dataset structure. Section V outlines the model training process and evaluation metrics, with the results presented and discussed in Section VI; final considerations and future objectives conclude the work in Section VII.

II. RELATED WORKS

In recent years, traditional orientation estimation techniques based on inertial measurements have significantly improved due to the incorporation of additional sensor data. This multi-sensor approach helps mitigate the error accumulation typical of AHRS systems and enhances their robustness.

As previously mentioned, one of the most common integration methods involves leveraging visual data, which is not only cost-effective but also rich in useful features. Therefore, the following subsections provide a brief overview of the existing literature in orientation estimation, beginning with traditional inertial-based methods before introducing vision-based approaches.

A. INERTIAL-BASED METHODS

Many studies have explored the use of inertial sensors for orientation estimation across various applications, including robotics [11], [12], human motion tracking [13], biologging for animal behavior research [14], aerial vehicles, aerospace [15], [16], gaming, virtual reality, and indoor pedestrian navigation [17], [18], [19].

Inertial sensors are widely favored due to their robust algorithms and high-accuracy results. Even relatively simple algorithms for position and orientation estimation are effective in practice, although the choice of the model can significantly affect overall accuracy [20]. Originally introduced in navigation systems, accurate inertial sensors and magnetic compasses are now commonly found in

consumer electronics, game consoles, and virtual reality devices. Nevertheless, challenges persist in representing orientation and fusing sensor data effectively [21].

Several studies have investigated real-time orientation estimation algorithms using low-cost IMUs. These range from earlier methods based on the Extended Kalman Filter [22] and complementary filter [23] to more recent applications for smartphone AHRS [24], [25]. Other work focuses on calibration methods for MEMS IMUs [26] and advanced denoising techniques using Deep Learning [27]. For example, Laidig et al. [28] proposed a filtering technique for acceleration measurements in a nearly inertial frame, achieving impressive results on public datasets. Similarly, Sun et al. [29] developed an algorithm that decouples pitch and roll estimates from magnetically disturbed environments, delivering superior performance across several experiments.

Additionally, the integration of neural networks with IMU-based data is gaining traction. Choi et al. [30] proposed an end-to-end recurrent neural network for attitude and heading estimation under various disturbance conditions, while Li et al. [31] combined a long short-term memory network and a Gauss–Markov model to reduce the impact of linear acceleration and magnetic disturbances. Seo et al. introduced DO IONet [32], a novel inertial odometry framework that minimizes drift errors through direct orientation estimation using inertial, gravitational, and geomagnetic data. Unlike existing methods, DO IONet estimates six degrees of freedom without initial values or cumulative errors, even over extended periods.

B. VISION-BASED METHODS

The use of visual data for orientation and pose estimation has garnered considerable attention over the past few decades. Many studies focus on horizon line detection, a crucial task for applications such as visual geo-localization and port security. However, marine environments present unique challenges, including interference from various factors. Wang et al. [33] developed a Sea-Sky Line (SSL) detection method for Unmanned Surface Vehicles (USVs) based on gradient saliency computation, while Jeong et al. [34] introduced a fast horizon line detection method for maritime scenarios using multi-scale approaches and region-of-interest (ROI) detection.

Horizon detection is also critical for unmanned aerial navigation. Carrio et al. [35] developed attitude estimation methods for thermal images using horizon line fitting and Convolutional Neural Networks (CNNs). Yoon et al. introduced MODAN [36], a multifocal object detection network for maritime horizon surveillance, using color quantization and ROI selection. Zardoua et al. [37] provided a comprehensive survey of horizon detection techniques.

A major challenge for vision-based methods lies in accounting for camera intrinsic and extrinsic parameters. To address this, several studies integrate visual, inertial, and magnetic data using Extended Kalman Filters (EKFs) [38], [39]. Visual Odometry (VO), Visual-Inertial Odometry (VIO), and Simultaneous Localization and Mapping (SLAM) algorithms are particularly popular for efficient ego-motion estimation in robotics [40], [41], [42], leveraging Deep Learning architectures such as LSTMs [43] and CNNs [44]. Mokssit et al. [45] conducted a survey on recent Deep Learning applications for SLAM.

Dual-camera setups have also been explored in UAV attitude estimation. Moore et al. [46] proposed a system using wide-angle imagery and fuzzy classification to determine the 3-DOF attitude of aircraft, outperforming an inertial unit in flight tests, though it was sensitive to lighting conditions. Duan et al. [47] improved visual odometry for UAV indoor navigation by rejecting outliers in stereo camera data. Teed et al. [48] developed Deep Patch Visual Odometry, a recurrent neural network for tracking image patches over time. Fu et al. [49] combined a hyper-Laplace filter with a CNN to enhance horizon line detection in the infrared spectrum.

A detailed review of Deep Learning models for inertial positioning estimation can be found in Chen et al. [50]. However, most of the literature focuses on stereo cameras [51], [52], [53] for orientation estimation and visual odometry. To the best of our knowledge, DC-DOES is the first approach that leverages a dual-camera system in a non-stereo configuration to develop a vision-based orientation estimation solution.

III. METHOD

This study presents DC-DOES, a hardware and software system for attitude estimation that leverages *dual-camera* data and a fully customizable Linux embedded platform for enhanced performance. The system consists of a low-cost, low-power embedded hardware platform equipped with an AHRS and a synchronized dual RGB camera setup. From a software perspective, DC-DOES has been developed through two sequential phases: first, training is performed using the previously acquired dataset DC-ROPIS, consisting of synchronized dual-camera images and corresponding AHRS measurements (used as ground truth). Next, a Deep Learning-based algorithm processes the image pairs from the two cameras to estimate roll and pitch angles.

DC-DOES is an improved version of DOES [10], which employed a low-cost monocular smartphone system to estimate orientation by analyzing the horizon at sea. In the previous version, the ROPIS dataset was collected using the FrameWOAndroid application, and training was carried out on a high-performance workstation.

This section provides an overview of the orientation estimation process (Sec. III-A), followed by a description of the deep neural network architectures selected as backbones for DC-DOES (Sec. III-B).

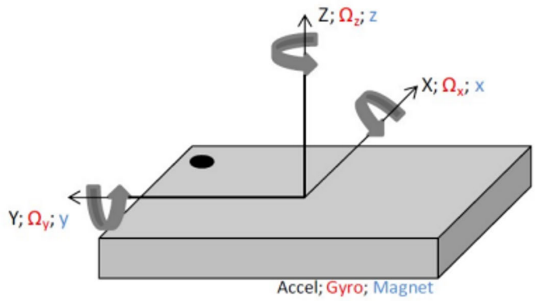


FIGURE 1. Coordinate frame definition of the BNO055 IMU [55].

A. THEORETICAL BACKGROUND ON ORIENTATION ESTIMATION

Transformation matrices are commonly used to define the orientation of a rigid body [54]. Among the available parameterizations, Euler angles are adopted in this work due to their intuitive representation of rotation around the principal axes of the body. The three Euler angles are defined as follows:

- *Roll* angle ϕ: rotation around the X axis.
- *Pitch* angle θ: rotation around the Y axis.
- *Yaw* angle ψ: rotation around the Z axis.

Figure 1 illustrates the coordinate frame of the AHRS device (IMU BNO055) used in this study.

To derive Euler angles, one can integrate raw measurements from a gyroscope, accelerometer, and magnetometer provided by an IMU or AHRS device. While angular velocity from the gyroscope can theoretically provide direct rotation angles through integration, this method often leads to accumulated errors over time (*gyroscope drift*). The accelerometer measures total acceleration, including gravity, which can be used for pitch and roll estimation, though it's highly sensitive to noise. The magnetometer, by providing a reference for the Earth's magnetic field, helps determine the yaw angle. Integrating these sensors' measurements mitigates individual sensor limitations, forming a standard approach for accurate orientation estimation.

B. DEEP ORIENTATION ESTIMATION

DC-DOES is a Deep Learning-based orientation estimation algorithm composed of a backbone neural network and two fully connected (FC) layers, designed to output roll and pitch estimates in parallel. The neural network processes two input images, positioned side by side along the horizontal axis. These images are resized to a resolution of 224 × 224 pixels, which is suitable for the selected backbone networks. Training is conducted using AHRS Euler angles as ground truth, with the corresponding image pairs serving as input in a regression task where roll and pitch are predicted as real-valued quantities.

A single backbone network is employed, while two separate fully connected layers are added after the last feature layer to perform distinct regression tasks for roll and pitch estimation. The deep models selected as backbones for DC-DOESinclude MaxViT, DenseNet161, ResNet18, and MobileNet V3.

MaxViT [56], based on the Vision Transformer (ViT) [57] architecture, introduces multi-axis parallelism, processing the input image across multiple sequences along different axes. This design leads to significant computational speedups. MaxViT also reduces the input image resolution using pooling and projection operations while preserving essential features, and it integrates an efficient attention mechanism for faster training and inference.

ResNet is a widely-used family of convolutional models that employ a *residual* architecture, which consists of residual blocks. In these blocks, feature maps produced by convolutional layers are combined with the input to compute an update (or residual) to the input feature maps. This structure addresses the vanishing gradient problem [58], improving both convergence speed and final accuracy. The lightweight ResNet18 model is selected for its efficiency and strong overall performance.

DenseNet [59], unlike ResNet, employs concatenation in its identity mappings to more efficiently preserve information. Each layer receives feature maps from all preceding layers, concatenates them, and passes the result through a non-linear transformation. This approach enhances feature reuse and improves gradient flow throughout the network. For this study, DenseNet161 was tested to evaluate DC-DOES's performance with a deeper, more computationally intensive network.

MobileNet V3 [60] is a state-of-the-art convolutional neural network designed for efficient performance on mobile and embedded devices. In its V3 iteration, MobileNet incorporates hardware-aware network architecture search, leveraging the NetAdapt [61] algorithm to optimize functionality. Key improvements include the hard swish activation function and squeeze-and-excitation modules within MBConv blocks, providing enhanced efficiency compared to earlier versions.

The backbone network and the fully connected layers are trained jointly using back-propagation with a Mean Square Error (MSE) loss function. Separate losses are computed for roll (L_{roll}) and pitch (L_{pitch}) as shown in Equations (1) and (2), where y represents the ground truth values and $\hat{y}$ the predicted values. The final loss, L_{final}, is the sum of these two losses, as shown in Equation (3).

$$L_{roll}(y_{roll}, \hat{y}_{roll}) = \frac{1}{n}\sum_{i=1}^{n}(y_{roll} - \hat{y}^i_{roll})^2 \tag{1}$$

$$L_{pitch}(y_{pitch}, \hat{y}_{pitch}) = \frac{1}{n}\sum_{i=1}^{n}(y_{pitch} - \hat{y}^i_{pitch})^2 \tag{2}$$

$$L_{final} = L_{roll} + L_{pitch} \tag{3}$$

All models are pre-trained on the ImageNet 1K dataset [62], allowing for the fine-tuning of pre-trained features to adapt them for the task at hand.

IV. ACQUISITION SYSTEM AND DATASET

This section presents the key components of the acquisition system (Sec. IV-A) and the new dataset, DC-ROPIS , created to train DC-DOES (Sec. IV-B). As discussed earlier, DC-DOES required a new dataset comprising paired images with environmental characteristics similar to the original ROPIS dataset to ensure a fair comparison. Additionally, using a custom, low-power Linux platform provided greater flexibility and customizability compared to the original Android platform. Therefore, a dual-camera smartphone was not used, and instead, a custom embedded system was built. This system consists of an Arducam 12MP MINI IMX477 Synchronized Stereo Camera mounted on a Nvidia Jetson Nano, with a BNO055 AHRS for ground-truth orientation measurements. The entire setup was enclosed in a 3D-printed chassis, as shown in Figure 2.

A. DEVICE INTERNAL SENSORS AND CHARACTERISTICS

One of the main challenges in creating a dual-camera dataset is ensuring the simultaneous capture of frames from both cameras. To address this, the Arducam 12MP MINI IMX477 Synchronized Stereo Camera Bundle Kit was used, allowing for simultaneous frame acquisition without delay when connected to the Jetson Nano platform. This bundle includes two high-quality cameras with a 1/2.3'' 12.3 Megapixel IMX477 sensor, offering a maximum resolution of 4056(H) x 3040(V) and a pixel size of $1.55\mu m$.

Each camera is equipped with an M12-Mount lens, featuring a manual focus ring and adjustable aperture. The lenses have a focal length of 3.9mm, a maximum aperture of 2.8, and a 75° horizontal field of view (FOV). Table 1 summarizes the main specifications of the cameras and lenses, as provided by the manufacturer [63].

Additionally, the camera baseline is adjustable, offering flexibility for use with different devices and enabling dynamic adjustments for a stereo binocular vision system.

In DC-DOES, the two cameras are positioned for a composite perspective: one camera faces forward, while the other is rotated 90 deg to the left, aligning with the X and Y axes of the AHRS sensors (Figure 2a). The Arducam stereo camera HAT facilitates the simultaneous operation of both cameras via a single MIPI CSI-2 connection, using ArduChip to present the dual-camera setup as a single camera to the Jetson Nano.

For inertial measurements, the low-cost BNO055 sensor by Bosch was chosen. This smart sensor integrates a triaxial 14-bit accelerometer, a triaxial 16-bit gyroscope, a triaxial geomagnetic sensor, and a 32-bit microcontroller running the BSX3.0 FusionLib software, all within a System in Package (SiP). The integration of these sensors with built-in sensor fusion simplifies its usage, while offering different configuration options. Detailed specifications are available in the official datasheet.[1] The AHRS was securely mounted on a breadboard to ensure precise alignment with the cameras' X and Y axes. The breadboard and camera HAT were then connected to the Jetson Nano (Figure 2b).

The Nvidia Jetson Nano is a system-on-module featuring a Maxwell™GPU with 128 CUDA cores and a Quad-Core ARM Cortex-A57 CPU, supported by 4GB of 64-bit memory. The Nvidia development kit[2] provides support for various APIs and Deep Learning frameworks such as PyTorch, TensorFlow, and ONNX. The Jetson Nano offers a 40-pin expansion header, a Micro-USB port for power input, a Gigabit Ethernet port, four USB 3.0 ports, an HDMI output, and two MIPI CSI-2 camera connectors. The system is powered by a standard smartphone power bank connected via the Micro-USB port, making it portable and deployable in diverse scenarios. A Samsung Evo Plus 64GB microSD card[3] with read/write speeds of 100 MB/s and 60 MB/s respectively, was used for booting the system and primary storage.

To protect the embedded system, a 3D-printed PLA case was created using a Creality Ender 3 printer.[4] The 3D model, designed using the AUTODESK Thinkercad webapp,[5] was specifically crafted to ensure the secure placement of all components and maintain precise sensor alignment. While slight misalignments between the camera and AHRS are not detrimental to DL-based methods due to their ability to implicitly learn camera rotation associations [64], a rigid structure ensures consistency throughout training and testing.

For usability, a touchscreen and an integrated mouse-keyboard device were connected to the Jetson Nano. The chosen touchscreen was a 121.11 × 95.24 mm Jun-Saxifragelec capacitive display with an 800 × 480 resolution, compatible with multiple platforms (Raspberry Pi, Windows, Ubuntu, Mac), supporting Raspbian, Ubuntu, and WIN11 IOT systems, and offering a framerate of up to 80fps. The screen connects to the Jetson Nano via HDMI and USB ports, with external power provided through the USB port. The keyboard-mouse combo, a Rii Mini i8,[6] has a QWERTY layout, a 1000DPI resolution touchpad, and wireless connectivity with a range of up to 10m (RF 2.4 GHz) and 8m (Bluetooth 4.0) (Figure 2a).

B. DATASET

To train the Deep Learning model, the dataset must contain a large volume of horizon images with the new dual-camera perspective, accompanied by precise ground truth (GT) data for roll and pitch angles. High accuracy in GT data is crucial, as it directly influences the learning process, depending on the instrumentation used (i.e., the BNO055 IMU).

A Python script has been developed to synchronize the Arducam stereo camera with the inertial measurement unit.

[1] https://www.bosch-sensortec.com/products/smart-sensor-systems/bno055/

[2] https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit

[3] https://www.samsung.com/it/memory-storage/memory-card/evo-plus-64gb-sd-card-2021-mb-sc64k-eu/

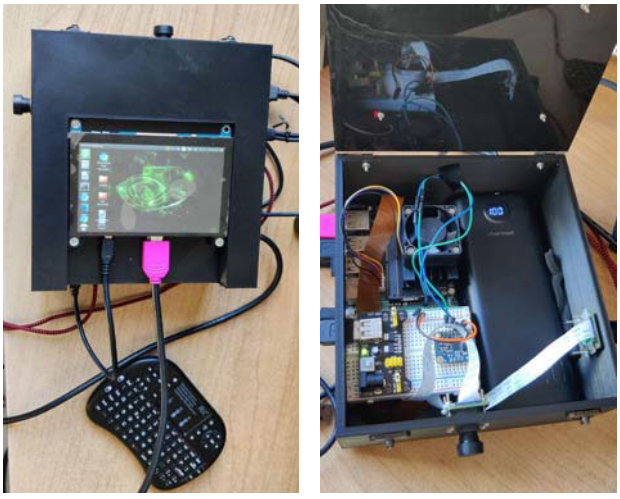
[4] https://www.creality.com/products/ender-3-3d-printer

[5] https://www.tinkercad.com/

[6] http://www.riitek.eu/IT/Prodotti/RT-MWK08+_IT.html

TABLE 1. Key specifications of the Arducam 12MP MINI IMX477 Synchronized Stereo Camera Bundle Kit, sensor and lens [63].

Camera	Lens
Sensor: IMX477	Format: 1/2.3 inch
Optical Format: 1/2.3" (diagonal 7.857mm)	Focal Length: 3.9mm
Resolution: 4056(H) x 3040(V) 12.3MP	Aperture (F): Max. F2.8
Pixel Size: 1.55um x 1.55um	Field of View (FOV): 75° (H)
IR Sensitivity: Visible light	Back Focal Length: 4.49mm
Interface: 4-lane MIPI CSI-2	MOD: 0.1m
Video Modes: 4056×3040@10fps, 2028×1529@40fps, 2028×1080@50fps	Dimension: 14×18.67mm
Board Size: 24mm × 25mm	Weight: 6g



(a) External view of the embedded system: the two cameras are placed at 90° one from the other, pointing to the front and to the left view respectively.

(b) Internal setup of the embedded system used to acquire the DC-ROPIS dataset and to further test DC-DOES; the breadboard with the wires connecting the IMU to the Jetson Nano, the camera HAT, and the Power Bank for the power supply can be seen in this picture.

FIGURE 2. Embedded system configuration for the deployment of DC-DOES and its 3D printed case.

Using the OpenCV[7] and Bosch BNO055 libraries,[8] the script efficiently retrieves visual and sensor data, ensuring minimal overhead and excellent synchronization between devices. The complete code is publicly available on GitHub at this link.

The BNO055 device can sample data at several hundred hertz, depending on the acquisition mode. However, the bottleneck in data collection arises during the reading of RGB data and the writing process to disk. To prevent delays and their cumulative effect on data acquisition, the rate has been empirically set to 7 fps. This ensures synchronized storage of both the images and the GT data, with the latter being recorded immediately after the RGB data acquisition. The GT data is provided directly by the internal integration of the sensor signals, such as acceleration, rotation, and magnetic field strength, using the $NDOF_FMC_OFF$ mode. This sensor fusion mode offers Nine Degrees of Freedom (NDOF) but disables Fast Magnetic Calibration (FMC), requiring manual calibration through a 'figure 8' pattern as suggested by the manufacturer. This calibration process is performed at the start of every acquisition to ensure reliable data, and the results are saved in a *calibration_data.txt* file for reference.

Once configured, the sensor provides fusion results like quaternions, Euler angles, linear acceleration, and gravity vectors at a fixed output rate. These values are accessible through I2C, UART, or HID-over-I2C interfaces. For the acquisition process, the default sensor axes orientation has been adopted, corresponding to the Windows format, with pitch values increasing with clockwise rotations (see Figure 1).

The DC-ROPIS dataset was acquired in Gaeta (Lazio) and Mondragone (Campania), Italy. It comprises 16,501 sRGB TrueColor JPEG images at a resolution of 1920 × 1080, totaling 15 GB, and is organized into 12 subdirectories. The data was collected in various locations, each presenting unique environmental factors, geographic characteristics, and weather-marine conditions. To ensure robust training, data was gathered in adverse weather conditions and low-light settings. Additionally, at least one camera in the dual setup faced partial occlusion in each scene, mimicking potential real-world visual challenges onboard.

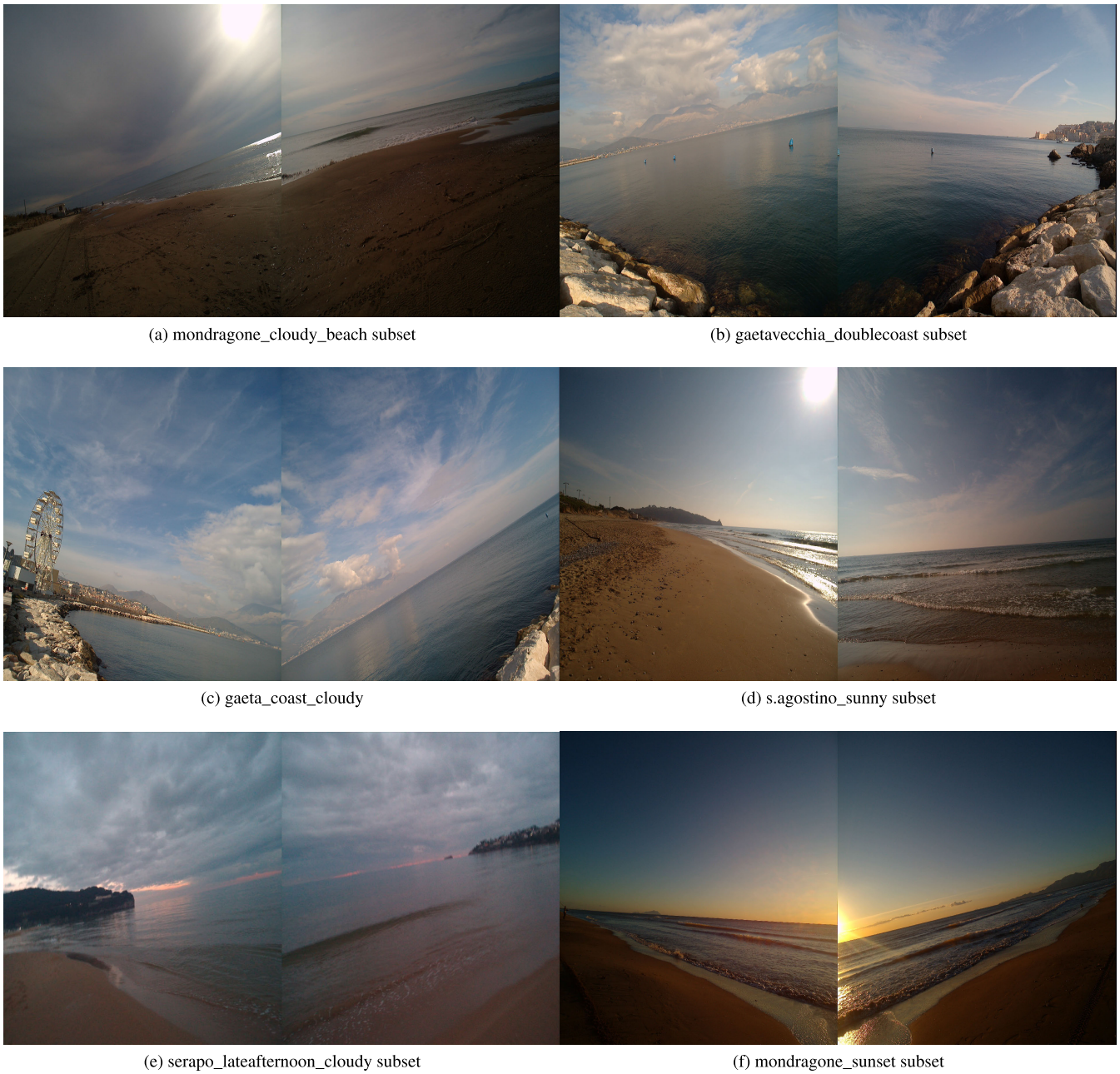
For training, 10 of the 12 acquisitions (13,432 images) were used, while the remaining two (3,069 images) were reserved for testing. The validation set, consisting of 15% of the training set, includes 2,370 images. Figure 3 showcases different samples from the DC-ROPIS dataset.

The inclusion of a dedicated test set composed of images from separate acquisitions allowed the evaluation of DC-DOES 's generalization capability in new scenes. Each acquisition was designed to simulate the behavior of a ship in navigation, under both static and dynamic conditions, replicating oscillations that mimic real ship movements.

It is important to highlight several key aspects of this dataset:

- The perspective of DC-ROPIS images, though slightly different from those captured onboard, includes foreground elements such as sand and rocks. However, this does not hinder the learning process, as Deep Learning models can differentiate between relevant and irrelevant image features.
- A true frame from a navigating vehicle would include visual elements like bow structures and parts of the bridge floor (or USV sections). While these specific

[7]https://opencv.org/
[8]https://github.com/boschsensortec/BNO055_SensorAPI



(a) mondragone_cloudy_beach subset

(b) gaetavecchia_doublecoast subset

(c) gaeta_coast_cloudy

(d) s.agostino_sunny subset

(e) serapo_lateafternoon_cloudy subset

(f) mondragone_sunset subset

FIGURE 3. **Examples of DC-ROPIS dataset samples acquired in different settings.**

features are absent in DC-ROPIS, DC-DOES has demonstrated robustness in handling similar visual clutter. Further experiments will assess the precise impact of such elements on learning.

- The camera was positioned at a consistent height of around 1.5 meters, with minimal vertical oscillations (heave) to simulate vehicle movements. Given that pitch estimation is closely tied to the horizon height, maintaining the horizon within the frame is critical.

Future enhancements to the DC-ROPIS dataset will involve acquiring data at various camera heights to analyze their impact on model training. Additionally, acquisitions will be expanded to more diverse scenarios, including adverse weather conditions and platforms such as ship bridges and USV platforms. The increased heterogeneity of data will improve the model's ability to generalize to complex, real-world environments.

V. EXPERIMENTAL SETUP

This section details the training process and provides a concise overview of the evaluation metrics used to assess the performance of DC-DOES.

FIGURE 4. Example of an image from the additional subset collected in Gaeta, which has been acquired under low-light and cloudy conditions. This challenging environment results in images with significant pixel noise.

A. TRAINING DETAILS

The work on DC-DOES has been entirely performed using Python as the programming language, employing PyTorch as the Deep Learning library. The code is publicly available.[9] Standard fine-tuning procedures were followed for the training, with the backbone convolutional kernels pre-trained on ImageNet, while the last fully connected layer was replaced with one with randomly initialized weights. The Adam optimizer was employed for fine-tuning the final model on the DC-ROPIS, with a learning rate fixed at 0.001. The number of training epochs was set to 20, as a larger number of epochs empirically corresponded to lower performance due to overfitting. As previously mentioned, the images were resized to 224 × 224 resolution, applying zero-padding to obtain a squared input without loss of information. A standard normalization procedure to zero mean-unit variance was applied to both the input images and the ground truth data, with normalization parameters calculated over the entire training set. The data augmentation process involved the application of random color variations to the images using PyTorch's *ColorJitter* transformation function. This function adjusts brightness, contrast, saturation, and hue by specified amounts, leading to an expanded training dataset that improved the generalization capabilities of DC-DOES. Due to the unique characteristics of the paired images, no random crop function was applied. The data augmentation procedure was disabled during the testing phase, while the zero-padding and resizing processes were still applied to the test images. Additionally, the predicted roll and pitch values were de-normalized before calculating the evaluation metrics discussed in the next section. Figure 5 visually summarizes the workflow of DC-DOES in its three main phases: data acquisition, training (including the specific data augmentation process), and the test phase, with the final performance evaluation.

Due to the absence of orientation estimation methods that utilize images acquired at a 90-degree angle, a direct comparison with other state-of-the-art solutions would be unfair. Deep learning models are typically trained on specific datasets, and without a comparable dataset of similarly angled images, benchmarking would not yield valid results. This limitation underscores the novelty of our approach while also highlighting an opportunity for future research in developing standardized datasets for orientation estimation at various angles.

[9]https://github.com/engharat/does2

B. EVALUATION METRICS

The performance of DC-DOES has been evaluated using standard regression metrics commonly employed in the literature.

The Mean Absolute Error (MAE), defined in Equation 4, is a risk metric that represents the expected value of the absolute error. It calculates the average absolute difference between predicted and actual values, maintaining the same scale as the data being measured. In MAE, each error contributes proportionally to its absolute value.

$$MAE(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

The Root Mean Square Error (RMSE) is the square root of the average of squared differences between predicted and observed values, also known as the quadratic mean of these differences (residuals). RMSE serves as an accuracy measure with particular sensitivity to outliers. By squaring errors before averaging, RMSE gives more weight to larger errors, making it especially useful when large errors are particularly undesirable. It's important to note that RMSE doesn't necessarily increase with error variance but grows with the variance in the frequency distribution of error magnitudes. Equation 5 presents the standard formulation of RMSE.

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

The Standard Deviation (STD) quantifies the dispersion or variation within a set of samples. A low standard deviation indicates that values cluster closely around the mean (or expected value), while a high standard deviation suggests a wider spread of values. Equation 6 presents the formula for standard deviation.

$$\sigma(\hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \mu)^2} \tag{6}$$

where μ represents the mean of the sample.

These three metrics have been calculated using the Scikit-learn library, specifically the *sklearn.metrics* module, which provides a comprehensive set of utility functions for measuring performance in regression tasks.

VI. RESULTS AND DISCUSSION

This section presents an analysis of the results obtained by DC-DOES on the considered dataset.

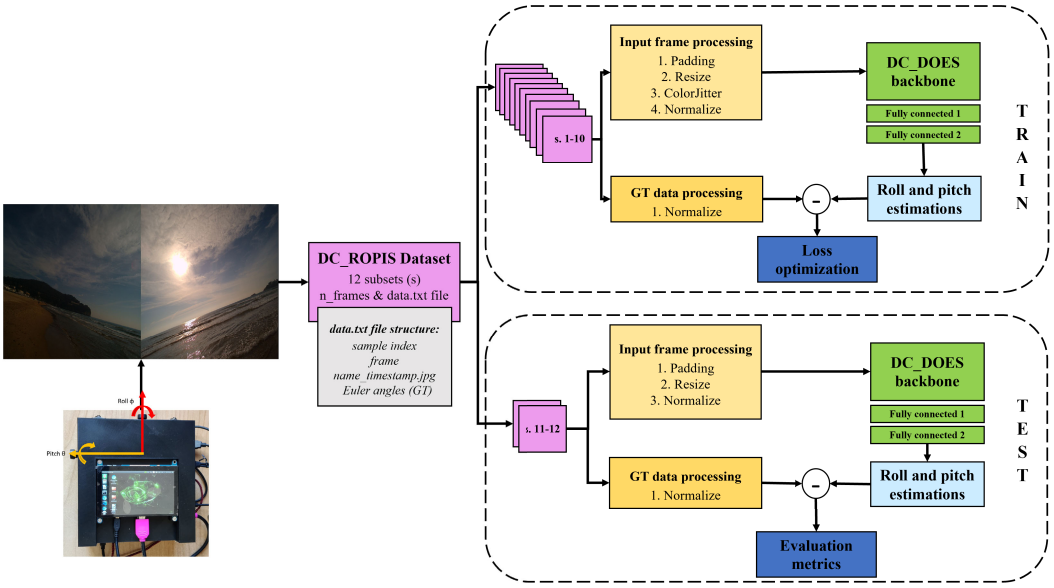


FIGURE 5. DC-DOES working flow: the data acquisition and dataset creation, the training phase with the applied data augmentation procedures, and the final test phase with the computation of the evaluation metrics.

To better interpret the results, it is important to understand the relationship between MAE and RMSE. These metrics trace error variation in predictions: RMSE is typically higher than MAE, and a larger difference indicates greater variance in individual sample errors. Conversely, when RMSE is close to MAE, it suggests that all errors have approximately the same magnitude. The STD values provide insight into the distribution of results relative to the mean. Smaller STD values indicate higher clustering of results around the mean, suggesting greater reliability. Due to RMSE's sensitivity to outliers, MAE has been selected as the primary metric for its robustness and ease of interpretation.

Table 2 displays DC-DOES performances with various backbone networks. The results demonstrate excellent performance for both roll and pitch angles, with the best-tested backbone achieving a Mean Absolute Error close to 1.5° for roll and nearly 1° for pitch.

All considered backbones produce satisfactory results, but the transformer model (MaxVit) demonstrates superior performance, emerging as the most suitable architecture for this task. The error reduction produced by MaxVit, while tangible, is relatively small compared to other tested convolutional networks.

A closer examination reveals that the performance gap between backbones is narrower for the pitch angle, with MAE values ranging from 1.17° (MaxVit) to 1.38° (MobileNetV3). The difference is more pronounced for the roll angle, where MaxVit achieves a MAE of 1.54°, while MobileNet reaches 1.80°.

The relatively poorer performance of MobileNet likely stems from its reduced model capacity, with only 4 million trainable parameters. However, it's noteworthy that MobileNet can run on low-powered embedded devices [9], producing a mean absolute error below 2 degrees for both roll and pitch angles—a satisfactory result for many applications.

Given these findings, the selection of DC-DOES backbones should be guided by the specific application requirements. MaxVit is the optimal choice for use with high-powered devices, while MobileNet is better suited for deployment on low-powered systems. ResNet18 provides a balanced compromise between performance and inference speed, with only a minor increase in MAE (0.16° for roll and 0.04° for pitch) compared to MaxVit. Notably, despite having a significantly higher number of trainable parameters, DenseNet161 does not outperform ResNet18 in this task.

MaxVit's superior performance is further confirmed by RMSE analysis: roll is estimated with an RMSE below 2 degrees, and pitch with an RMSE even lower than 1.60 degrees. The RMSE results of other networks follow a similar pattern to MAE, with ResNet18 and DenseNet161 showing comparable accuracy, while MobileNet produces the highest RMSE at 2.31 and 1.86 degrees for roll and pitch, respectively.

Moreover, all four networks exhibit small STD values, indicating uniform error distribution across the dataset, with few samples producing high estimation errors. This trend aligns with previous metrics, MaxVit showing the smallest STD values for roll, while ResNet18 surprisingly performs best for pitch.

Table 2 also presents inference speeds, measured as the time required for a single roll and pitch angle prediction from one image pair, excluding image loading and preprocessing. This metric is particularly important for deploying the

TABLE 2. Comparative results on different DC-DOES backbones. *TP* indicates the number of trainable parameters.

	MaxViT $TP = 30M$ t = 56 msec		ResNet18 $TP = 11M$ t = 11 msec		MobileNet $TP = 4M$ t = 4msec		DenseNet161 $TP = 26M$ t = 35 msec	
	roll	*pitch*	*roll*	*pitch*	*roll*	*pitch*	*roll*	*pitch*
MAE [deg]	**1.54**	**1.17**	1.70	1.21	1.80	1.38	1.68	1.22
RMSE [deg]	1.97	1.60	2.17	1.54	2.31	1.86	2.21	1.59
STD [deg]	1.22	1.09	1.35	0.95	1.45	1.24	1.30	1.01

algorithm on devices where speed is a critical factor, such as in real-time applications, typically defined as 30 fps.

In terms of the speed-accuracy trade-off, MobileNet delivers the fastest inference at 4 milliseconds while keeping the MAE below 2 degrees. MaxVit, though the most accurate, is the slowest with an inference time of 56 milliseconds, making it unsuitable for real-time predictions at 30 fps. ResNet18 strikes a balance between accuracy and speed, with an inference time of 11 milliseconds. It should be noted that these timings can vary depending on the hardware used for testing.

In summary, MaxVit is recommended when a dedicated GPU is available, offering high-speed inference with best estimation performance. For low-powered, embedded devices, ResNet18 demonstrates good accuracy with lower computational complexity, making it suitable for constrained environments.

A comparison between DC-DOES and its predecessor, DOES [10], reveals sensible improvements. The roll MAE has improved by 0.11 degrees, while the pitch MAE shows a significant improvement of 0.67 degrees, when comparing the best-performing backbones. Using the same backbone, ResNet18, DC-DOES maintains the roll performance of DOES but substantially enhances pitch prediction, attributed to the simultaneous use of two images. However, this comparison remains task-specific due to differences in the test datasets.

To assess the robustness of DC-DOES under challenging conditions, an additional set of 1044 images with significant noise was collected. As shown in Figure 4, these images were captured at sunset with a cloudy sky and very low light. Despite these adverse conditions, DC-DOES achieves an MAE of 2.62 for roll angle and 1.98 for pitch angle, demonstrating its robustness to varying lighting conditions. Further data collection is ongoing to improve resilience across diverse environmental setting.

VII. CONCLUSION

This paper introduces DC-DOES, Double Camera - Deep Orientation (of roll and pitch) Estimation at Sea, a novel approach to enhancing orientation estimation using Deep Learning techniques and a dual-camera system. The study demonstrates significant advancements in the accuracy of roll and pitch angle estimations, which are essential for various applications beyond the maritime domain.

Key contributions include the development of a dual-camera embedded system integrated with an Nvidia Jetson Nano, which enabled the creation of the Double Camera - ROll and PItch (DC-ROPIS) dataset, tailored for orientation estimation tasks. A comparative evaluation of several Deep Learning backbone architectures, including MaxVit, MobileNetV3, ResNet18, and DenseNet161, showed that MaxVit achieves the highest accuracy, with a Mean Absolute Error (MAE) of 1.17° for pitch and 1.54° for roll angles. This underscores the potential of transformer models in improving the robustness of visual-based orientation estimation methods.

Despite MaxVit's superior performance, ResNet18 offers an attractive trade-off between accuracy and inference speed, making it suitable for deployment on computationally constrained embedded devices, such as those commonly used in robotics. While MobileNetV3 exhibited the highest MAE among the tested backbones, its fast inference time makes it a viable option for real-time processing applications.

These findings highlight the importance of selecting the appropriate backbone architecture based on specific application requirements, such as prioritizing high accuracy versus real-time performance. Additional experiments were also conducted on a newly acquired subset of data, which included noisy images and low-light conditions. The results further validated DC-DOES's robustness in challenging environments, demonstrating its ability to maintain accuracy under adverse conditions.

Future research will focus on refining the dual-camera system to differentiate between pitch rotation and vertical movement along the z-axis, potentially extending the system's applicability to other challenging scenarios. Moreover, integrating additional sensor modalities and leveraging advanced Deep Learning techniques could further enhance the system's accuracy and robustness. This will be supported by the collection of new datasets in more realistic environments, such as mounting the embedded system on robots, paving the way for the deployment of DC-DOES in real-time operational contexts.

In conclusion, this research contributes to the growing field of pose estimation by offering a cost-effective, accurate, and reliable approach to orientation estimation, with applications in various domains that require robust attitude estimation.

REFERENCES

[1] J. Melo and A. Matos, "Survey on advances on terrain based navigation for autonomous underwater vehicles," *Ocean Eng.*, vol. 139, pp. 250–264, Jul. 2017.

[2] L. Alessandri, V. Baiocchi, S. D. Pizzo, F. Di Ciaccio, M. Onori, M. F. Rolfo, and S. Troisi, "Three-dimensional survey of guattari cave with traditional and mobile phone cameras," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 42, pp. 37–41, May 2019.

[3] Y. Zhang, X. Chen, L. Wei, J. Che, P. Liu, and J.-H. Cui, "VISS-CF: Visual-inertial odometry and sonar fused SLAM framework with enhanced corner feature matching for underwater environment," in *Proc. 17th Int. Conf. Underwater Netw. Syst.*, Nov. 2023, pp. 1–5.

[4] F. Di Ciaccio and S. Troisi, "Monitoring marine environments with autonomous underwater vehicles: A bibliometric analysis," *Results Eng.*, vol. 9, Mar. 2021, Art. no. 100205.

[5] K. Xu, Y. Hao, S. Yuan, C. Wang, and L. Xie, "AirVO: An illumination-robust point-line visual odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 3429–3436.

[6] W. Chen, G. Shang, A. Ji, C. Zhou, X. Wang, C. Xu, Z. Li, and K. Hu, "An overview on visual SLAM: From tradition to semantic," *Remote Sens.*, vol. 14, no. 13, p. 3010, Jun. 2022.

[7] H.-Y. Lin and J.-R. Zhan, "GNSS-denied UAV indoor navigation with UWB incorporated visual inertial odometry," *Measurement*, vol. 206, Jan. 2023, Art. no. 112256.

[8] D. Weber, C. Gühmann, and T. Seel, "RIANN—A robust neural network outperforms attitude estimation filters," *AI*, vol. 2, no. 3, pp. 444–463, Sep. 2021.

[9] P. Russo and F. Di Ciaccio, "Deep models optimization on embedded devices to improve the orientation estimation task at sea," in *Proc. IEEE Int. Workshop Metrol. Sea; Learn. Measure Sea Health Parameters (MetroSea)*, vol. 29, Oct. 2022, pp. 44–49.

[10] F. D. Ciaccio, P. Russo, and S. Troisi, "DOES: A deep learning-based approach to estimate roll and pitch at sea," *IEEE Access*, vol. 10, pp. 29307–29321, 2022.

[11] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on various inertial measurement unit (IMU) sensor applications," *Int. J. Signal Process. Syst.*, vol. 1, no. 2, pp. 256–262, 2013.

[12] J. S. Lora-Millan, A. F. Hidalgo, and E. Rocon, "An IMUs-based extended Kalman filter to estimate gait lower limb sagittal kinematics for the control of wearable robotic devices," *IEEE Access*, vol. 9, pp. 144540–144554, 2021.

[13] A. Filippeschi, N. Schmitz, M. Miezal, G. Bleser, E. Ruffaldi, and D. Stricker, "Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion," *Sensors*, vol. 17, no. 6, p. 1257, Jun. 2017.

[14] H. J. Williams, L. A. Taylor, S. Benhamou, A. I. Bijleveld, T. A. Clay, S. de Grissac, U. Demšar, H. M. English, N. Franconi, A. Gómez-Laich, R. C. Griffiths, W. P. Kay, J. M. Morales, J. R. Potts, K. F. Rogerson, C. Rutz, A. Spelt, A. M. Trevail, R. P. Wilson, and L. Börger, "Optimizing the use of biologgers for movement ecology research," *J. Animal Ecol.*, vol. 89, no. 1, pp. 186–206, Jan. 2020.

[15] S. Adler, S. Schmitt, K. Wolter, and M. Kyas, "A survey of experimental evaluation in indoor localization research," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2015, pp. 1–10.

[16] H. G. de Marina, F. J. Pereda, J. M. Giron-Sierra, and F. Espinosa, "UAV attitude estimation using unscented Kalman filter and TRIAD," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4465–4474, Nov. 2012.

[17] E. Vertzberger and I. Klein, "Attitude adaptive estimation with smartphone classification for pedestrian navigation," *IEEE Sensors J.*, vol. 21, no. 7, pp. 9341–9348, Apr. 2021.

[18] V. Renaudin and C. Combettes, "Magnetic, acceleration fields and gyroscope quaternion (MAGYQ)-based attitude estimation with smartphone sensors for indoor pedestrian navigation," *Sensors*, vol. 14, no. 12, pp. 22864–22890, Dec. 2014.

[19] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1281–1293, 3rd Quart., 2013.

[20] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Found. Trends Signal Process.*, vol. 11, nos. 1–2, pp. 1–153, Nov. 2017, doi: 10.1561/2000000094. [Online]. Available: https://www.nowpublishers.com/article/Details/SIG-094

[21] N. H. Q. Phuong, H.-J. Kang, Y.-S. Suh, and Y.-S. Ro, "A DCM based orientation estimation algorithm with an inertial measurement unit and a magnetic compass," *J. Universal Comput. Sci.*, vol. 15, no. 4, pp. 859–876, 2009.

[22] A. Kim and M. F. Golnaraghi, "A quaternion-based orientation estimation algorithm using an inertial measurement unit," in *Proc. Position Location Navigat. Symp.*, Apr. 2004, pp. 268–272.

[23] R. Valenti, I. Dryanovski, and J. Xiao, "Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs," *Sensors*, vol. 15, no. 8, pp. 19302–19330, Aug. 2015.

[24] T. Michel, P. Genevès, H. Fourati, and N. Layaïda, "On attitude estimation with smartphones," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2017, pp. 267–275.

[25] F. Di Ciaccio, S. Gaglione, and S. Troisi, "A preliminary study on attitude measurement systems based on low cost sensors," in *Proc. Int. Workshop R3 Geomatics, Res., Results Rev.* Berlin, Germany: Springer, 2019, pp. 103–115.

[26] J. Schnee, J. Stegmaier, T. Lipowsky, and P. Li, "Auto-correction of 3D-orientation of IMUs on electric bicycles," *Sensors*, vol. 20, no. 3, p. 589, Jan. 2020.

[27] P. Russo, F. Di Ciaccio, and S. Troisi, "DANAE++: A smart approach for denoising underwater attitude estimation," *Sensors*, vol. 21, no. 4, p. 1526, Feb. 2021.

[28] D. Laidig and T. Seel, "VQF: Highly accurate IMU orientation estimation with bias estimation and magnetic disturbance rejection," *Inf. Fusion*, vol. 91, pp. 187–204, Mar. 2023.

[29] Y. Sun, X. Xu, X. Tian, L. Zhou, and Y. Li, "A quaternion-based sensor fusion approach using orthogonal observations from 9D inertial and magnetic information," *Inf. Fusion*, vol. 90, pp. 138–147, Feb. 2023.

[30] J. S. Choi, C. J. Lee, and J. K. Lee, "A parallel recurrent neural network for robust inertial and magnetic sensor-based 3D orientation estimation," *IEEE Access*, vol. 11, pp. 89685–89693, 2023.

[31] P. Li, W.-A. Zhang, Y. Jin, Z. Hu, and L. Wang, "Attitude estimation using iterative indirect Kalman with neural network for inertial sensors," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–10, 2023.

[32] H.-I. Seo, J.-W. Bae, W.-Y. Kim, and D.-H. Seo, "DO IONet: 9-axis IMU-based 6-DOF odometry framework using neural network for direct orientation estimation," *IEEE Access*, vol. 11, pp. 55380–55388, 2023.

[33] B. Wang, Y. Su, and L. Wan, "A sea-sky line detection method for unmanned surface vehicles based on gradient saliency," *Sensors*, vol. 16, no. 4, p. 543, Apr. 2016.

[34] C. Y. Jeong, H. S. Yang, and K. Moon, "Fast horizon detection in maritime images using region-of-interest," *Int. J. Distrib. Sensor Netw.*, vol. 14, no. 7, Jul. 2018, Art. no. 155014771879075.

[35] A. Carrio, H. Bavle, and P. Campoy, "Attitude estimation using horizon detection in thermal images," *Int. J. Micro Air Vehicles*, vol. 10, no. 4, pp. 352–361, Dec. 2018.

[36] S. Yoon, A. Jalal, and J. Cho, "MODAN: Multifocal object detection associative network for maritime horizon surveillance," *J. Mar. Sci. Eng.*, vol. 11, no. 10, p. 1890, Sep. 2023.

[37] Y. Zardoua, A. Astito, and M. Boulaala, "A survey on horizon detection algorithms for maritime video surveillance: Advances and future techniques," *Vis. Comput.*, vol. 39, no. 1, pp. 197–217, Jan. 2023.

[38] G. Ligorio and A. Sabatini, "Extended Kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: Comparative analysis and performance evaluation," *Sensors*, vol. 13, no. 2, pp. 1919–1941, Feb. 2013.

[39] M. Alatise and G. Hancke, "Pose estimation of a mobile robot based on fusion of IMU data and vision data using an extended Kalman filter," *Sensors*, vol. 17, no. 10, p. 2164, Sep. 2017.

[40] E. Hong and J. Lim, "Visual-inertial odometry with robust initialization and online scale estimation," *Sensors*, vol. 18, no. 12, p. 4287, Dec. 2018.

[41] T. Feng and D. Gu, "SGANVO: Unsupervised deep visual odometry and depth estimation with stacked generative adversarial networks," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4431–4437, Oct. 2019.

[42] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, "Monocular depth estimation based on deep learning: An overview," *Sci. China Technol. Sci.*, vol. 63, no. 9, pp. 1612–1627, Sep. 2020.

[43] J. R. Rambach, A. Tewari, A. Pagani, and D. Stricker, "Learning to fuse: A deep learning approach to visual-inertial camera pose estimation," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Sep. 2016, pp. 71–76.

[44] L. Han, Y. Lin, G. Du, and S. Lian, "DeepVIO: Self-supervised deep learning of monocular visual inertial odometry using 3D geometric constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6906–6913.

[45] S. Mokssit, D. B. Licea, B. Guermah, and M. Ghogho, "Deep learning techniques for visual SLAM: A survey," *IEEE Access*, vol. 11, pp. 20026–20050, 2023.

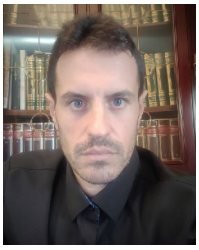
[46] R. J. Moore, S. Thurrowgood, D. Soccol, D. Bland, and M. V. Srinivasan, "A method for the visual estimation and control of 3-DOF attitude for UAVs," in *Proc. Australas. Conf. Robot. Autom. (ACRA)*, Melbourne, VIC, Australia, 2011, pp. 267–275.
[47] R. Duan, D. P. Paudel, C. Fu, and P. Lu, "Stereo orientation prior for UAV robust and accurate visual odometry," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 5, pp. 3440–3450, Oct. 2022.
[48] Z. Teed, L. Lipson, and J. Deng, "Deep patch visual odometry," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023
[49] J. Fu, F. Li, and J. Zhao, "Real-time infrared horizon detection in maritime and land environments based on hyper-Laplace filter and convolutional neural network," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.
[50] C. Chen and X. Pan, "Deep learning for inertial positioning: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 9, pp. 10506–10523, Sep. 2024.
[51] Z. Xu, M. Haroutunian, A. J. Murphy, J. Neasham, and R. Norman, "An integrated visual odometry system with stereo camera for unmanned underwater vehicles," *IEEE Access*, vol. 10, pp. 71329–71343, 2022.
[52] J. Zhang, Z. Liu, Y. Gao, and G. Zhang, "Robust method for measuring the position and orientation of drogue based on stereo vision," *IEEE Trans. Ind. Electron.*, vol. 68, no. 5, pp. 4298–4308, May 2021.
[53] P. Liu, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys, "Direct visual odometry for a fisheye-stereo camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1746–1752.
[54] P. Bernal-Polo and H. Martínez Barberá, "Orientation estimation by means of extended Kalman filter, quaternions, and charts," *J. Phys. Agents*, vol. 8, no. 1, pp. 11–24, 2017.
[55] Bosh. *BMI260: IMU Combining Accelerometer and Gyroscope*. Accessed: Jul. 1, 2024. [Online]. Available: https://www.bosch-sensortec.com/products/motion-sensors/imus/bmi260/
[56] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "MaxViT: Multi-axis vision transformer," in *Proc. 17th Eur. Conf. Comput. Vis.* Basel, Switzerland: Springer, 2022, pp. 459–479.
[57] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, Austria, May 2021, pp. 1–21. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy
[58] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 550–558.
[59] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[60] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.
[61] T.-J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, M. Sandler, V. Sze, and H. Adam, "NetAdapt: Platform-aware neural network adaptation for mobile applications," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 285–300.
[62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[63] Arducam. (2023). *Arducam 12MP MINI IMX477 Synchronized Stereo Camera Bundle Kit for Jetson Nano and Xavier NX*. [Online]. Available: https://www.arducam.com/product/arducam-12mp-mini-imx477-synchronized-stereo-camera-bu
[64] R. Zhu, M. Yang, W. Liu, R. Song, B. Yan, and Z. Xiao, "DeepAVO: Efficient pose refining with feature distilling for deep visual odometry," *Neurocomputing*, vol. 467, pp. 22–35, Jan. 2022.

FABIANA DI CIACCIO received the B.S. degree in nautical and aeronautical sciences and the master's degree (cum laude) in science and technology of navigation from the Parthenope University of Naples, Italy, in 2015 and 2018, respectively, and the Ph.D. degree from the "Environment, Resources and Sustainable Development" Program, UNESCO Chair, Parthenope University of Naples, in 2022.

She is an Assistant Professor (RTDa) with the Department of Civil and Environmental Engineering, University of Florence; and a member of the Geomatics for Environment and Conservation of Cultural Heritage (GECO) Laboratory. Currently, she teaches the course of remote sensing for the environmental engineering master's degree. Her research interests include attitude estimation methods based on visual-inertial systems, computer vision, and deep learning techniques, along with cultural heritage preservation, climate change impact assessment, environmental monitoring, metrology, underwater photogrammetry, 3-D reconstruction techniques, and AUV mapping.

SALVATORE TROISI received the degree (Hons.) in nautical sciences from the Faculty of Nautical Sciences, Naval University in Naples, in January 1984, with an experimental thesis in nautical astronomy on the use of optical amplification of light.

He has been a Full Professor SSD ICAR06 with the Faculty of Science and Technology, Parthenope University of Naples, since 2007. Since 1987, he has been a Researcher of the group 135 (first discipline complements topography), Faculty of Nautical Sciences, Naval University of Naples. From November 1998 to September 2007, he was an Associate Professor of SSD ICAR 06 with the Faculty of Nautical Sciences, Naval University of Naples. His scientific work regards both theoretical elaborations and measuring operations, regarding issues in the following areas: deformations control networks; geoid by astrogeodetic methods and GPS; topographic methods in environmental emergencies; GPS survey for deformations; design and simulation of geodetic networks by GPS methodology; design of satellite constellations; laser scanning; filtering of laser scanning data; close-range photogrammetry for reverse engineering; 3-D building modeling by aerial laser scanning data.



PAOLO RUSSO was born in Formia, Italy, in 1985. He received the B.S. degree in telecommunication engineering from the University of Cassino, in 2008, and the M.S. degree in artificial intelligence and robotics and the Ph.D. degree in computer science from Sapienza University of Rome, in 2016 and 2020, respectively.

He is currently an Assistant Professor with AlcorLab, DIAG Department, University of Rome Sapienza. From 2018 to 2019, he conducted research with the Italian Institute of Technology (IIT), Turin, Italy. He currently teaches interactive computer graphics. His research focuses on deep learning, computer vision, generative adversarial networks, and signal processing. Further information and a comprehensive list of his publications are available on his website at and https://www.paolorusso.org.

Open Access funding provided by 'Università degli Studi di Roma "La Sapienza" 2' within the CRUI CARE Agreement